Detailed Action

Remarks

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 07/14/2009 has been entered.
2. This office action is in response to the amendment filed on 07/14/2009.
3. Claims 2 and 20 have been cancelled.
4. Claims 22-24 have been added.
5. Claims 1, 18 and 21 have been amended
6. The objection to claims 1-4, 6-8 and 17 is withdrawn in view of Applicants' amendment.
7. Claims 1, 3, 4, 6-8, 17-19 and 21-24 remain pending and have been examined.
8. An examiner-initiated phone interview was conducted on 09/21/2009 before this Office Action to discuss and clarify the claim interpretation regarding to the specification disclosure. See Examiner-Initiated Interview Summary (Page No. 20090914) for details.

Response to Arguments

9. Applicant's arguments filed on 07/14/2009, in particular on pages 6-7, have been fully considered but are moot in view of new ground(s) of rejection applied in this Office action.

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

11. Claims 1, 7, 17-19, 21 and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Crump (Crump et al. US 5,850,56) in view of Sanchez (Sanchez et al., US 6,477,666) and in further view of Hundt (Robert Hundt, US 2004/0205720)

Claim 1:

Crump discloses a method for testing and debugging computer programs, the method comprising:

- Setting a plurality of breakpoints corresponding to a plurality of events in a Basic Input/Output System (BIOS) program code, each event being a test executed by the BIOS program code to a peripheral device (see for example ABSTRACT, "A monitor and debugger routine operable on a personal computer fro facilitating the design of power-on self test (POST) and basic

input and output system (BIOS) code."; also see col.11, lines 15-40, "Set Breakpoint command" and related text) or an error processing path when the peripheral device is in an error state (see for example, col.12, lines 5-49, "After a Go command is acted upon, the CPU 40 will execute the code indefinitely, until either an instruction breakpoint fault occurs or a data-breakpoint trap occurs", "As stated above, if either breakpoint is triggered, then execution control is transferred back to the monitor and debugger routine..." and related text);

- Executing the BIOS program code for outputting a diagnosis code of a breakpoint (see for example, col.12, lines 50-56, "...the monitor and debugger routine causes the external communications device to display the data byte located at desired I/O port" and related text);
- Setting a parameter (modify memory) (see for example, col.13, lines 6-30, "modify the data located at the desired region of memory" and related text);

But Crump does not explicitly disclose setting/resetting a parameter to simulate the peripheral device being in the error state throughout execution of the event corresponding to the diagnosis code, However, Sanchez in the same analogous art of testing the computer program about reliable and proper handling of various faults under various conditions, discloses a method to:

- simulating the error state throughout execution (injecting faults and error) (see for example, Fig.9, step 70, "Configure Program/Application for automatic fault injection by setting one or more breakpoints within the

program/application wherein the breakpoints are where faults may be injected"; step 72, "automatic fault injector is initiated" and related text).

- Resetting a parameter of the event corresponding to the diagnosis code (see for example, Fig.9, step 78 "Should a fault be inserted", step 80, "Pick one of the exceptions for this method and throw it" and related text);
- Executing the event corresponding to the diagnosis code according to the reset parameter for making the event undergo the error processing path (see for example, Fig.9, step 78 "Should a fault be inserted", step 80, "Pick one of the exceptions for this method and throw it" and related text);

Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use Sanchez's fault injection method to simulate the error state of peripheral device in Crump. One would have been motivated to do so to test the reliable and proper handling of various faults and exceptions under various conditions as suggested by Sanchez (see for example, ABSTRACT, lines 2-4, "to test the reliable and proper handling of various faults and exceptions under various conditions").

Neither Crump nor Sanchez explicitly discloses setting a parameter to simulate the peripheral device is working well throughout execution of the event corresponding to the diagnosis code and executing the event corresponding to the diagnosis code according to the parameter for making the event undergo the general processing path. However, Hundt in the same analogous art of testing / debugger a program discloses "At each breakpoint, the programmer examines

and changes the value of the program variable, redirects the program flow..." (see for example, paragraph [0002]); and "At each breakpoint, the program is stopped, a debugging prompt is provided to the user, and user enters debugging commands...The user then allows the program to continue execution" (see for example, paragraph [0003]). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to check and/or modify (set/reset) the parameters of the program to simulate different conditions including "error state" and/or "working well". One would have been motivated to do so to redirect the program execution flow as suggested by Hundt (see for example, paragraph [0002]) and test the reliable and proper handling of various faults and exceptions under various conditions as suggested by Sanchez (see for example, ABSTRACT, lines 2-4, "to test the reliable and proper handling of various faults and exceptions under various conditions")

Claim 7:

Crump, Sanchez and Hundt disclose the method for program debugging as in claim 1 above, Crump further discloses reset vector after the system power is applied or the system is reset. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was make to fully test all the error or fault handling as suggested by Sanchez (see for example, ABSTRACT, lines 2-4, "to test the reliable and proper handling of various faults and exceptions under various conditions"). Because reset procedures as one of

error handling feature is well known in the art as also indicated by Crump (see for example, col.7, lines 63, "Such reset procedures are well known in the art")

Claim 17:

Crump further discloses the method of claim 1 further comprising executing the BIOS program code until the diagnosis code of the breakpoint matches a predetermined diagnosis code before resetting the parameter of the event corresponding to the diagnosis code (see for example, col.16, lines 43-57, "... (4) examining relevant memory and registers using he display memory and... comparing the displayed results with the expected result..." and related text).

Claims 18-19 and 21-22:

Crump, Sanchez and Hundt disclose the same method for program debugging as addressed in Claim 1 above. Claims 18-19 and 21-22 are other version of claimed method as recited in claim 1. All the limitations have been disclosed by Crump, Sanchez and Hundt. Therefore, claims 18-19 and 21-22 also would have been obvious in view of reference teachings above.

Claim 24:

Crump, Sanchez and Hundt disclose the method of claim 22, Sanchez further discloses when executing the BIOS program code according to the reset parameter and the reset parameter determines the critical event error handling

path is to be taken, the critical error handling path generates an audible tone, a system reset, or a stop execution command (see for example, Fig.9, step 80, "Pick one of the exceptions for this method and throw it.")

12. Claims 3-4 and 8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Crump (Crump et al. US 5,463,766) in view of Sanchez (Sanchez et al., US 6,477,666) and Hundt (Robert Hundt, US 2004/0205720), and in further view of Phillips (Phillips et al., US 5,321,828)

Claim 3-4:

Crump discloses the method for program debugging as in claim 1 above, but does not explicitly disclose the breakpoints are set ahead of program codes of the corresponding events or after program codes of the corresponding events. However, Phillips in the same analogous art of an in-circuit emulator for hardware/software development and debugging microprocessors discloses that a user to set any number of breakpoints all at the same place in the program, or at different places (see for example, col.26-col.27, section "Setting Breakpoints" and related descriptions). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to set breakpoints anywhere in the code in order to adequately support execution control functionality and provide the rich set of functionality needed for the debugger. One would have been motivated to set breakpoints before or after the program

codes of the corresponding events to narrow down the places where the bugs might occur.

Claim 8:
Crump, Sanchez and Hundt disclose the method for program debugging as in claim 1 above which has an error handler to display error message, but do not explicitly disclose the error handler is a system execution interrupt. However, Phillips in the same analogous art of an in-circuit emulator for hardware/software development and debugging microprocessors discloses that execution interrupt (see for example, col.72, lines 60-67, "single interrupt request line"). Therefore, it would have been obvious to one having ordinary skill in the art at the time the inversion was make to use the method of system execution interrupt to allow the control processor to monitor the Clock Detect signals which is suggested by Phillips. One would have been motivated to do so to stop executing or suspend current process to trace the problem.

13. Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over Crump (Crump et al. US 5,838,975) in view of Sanchez (Sanchez et al., US 6,477,666) and Hundt (Robert Hundt, US 2004/0205720), and in further view of Robinson (Jeffrey I. Robinson, US 5,768,591)

Claim 6:
Crump, Sanchez and Hundt disclose the method for program debugging as in claim 1 above, but do not explicitly disclose that the error handler is an audible

tone. However, Robinson discloses a similar method for program debugging as in claim 1 above which the error handler is an audible tone. (Fig.4, items 172, 164, col.12, lines 64-67, "A sound generator 164 is provided and controlled by the message parser and error handler 172"). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use "sound generator" to replace Crump's method of error handler. One would have been motivated to do so to generate alarm to alert the user when the bug occurs.

14. Claim 23 is rejected under 35 U.S.C. 103(a) as being unpatentable over Crump (Crump et al. US 5,463,766) in view of Sanchez (Sanchez et al., US 6,477,666) and Hundt (Robert Hundt, US 2004/0205720), and in further view of Treu (Albert R. Treu, US 5,245,615)

Claim 23:

Crump, Sanchez and Hundt discloses the method of claim 22 comprising when executing the BIOS program code according to the reset parameter and the reset parameter determines the generic event error handling path is to be taken, but none of them explicitly discloses writing error messages to a file. However, Treu in same art discloses writing error message to a file (error log) (see for example, Fig.4, step 194-200, "Write Log Info In Error Log" and related text). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to do so for later accessing and diagnosing as suggested by

Treu (see for example, SUMMARY, "...can readily be later accessed after error logging has occurred...for use in logging errors and diagnosis of such errors")

Conclusion

15. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

16. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Zheng Wei whose telephone number is (571) 270-1059 and Fax number is (571) 270-2059. The examiner can normally be reached on Monday-Thursday 8:00-15:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature of relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571- 272-1000.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Z. W./
Examiner, Art Unit 2192

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192